ซอฟต์แวร์
ลออ เพิ่มสมบัติ

เริ่มต้นการเขียนโปรแกรม กราฟิก อย่างมีระบบ ตอนที่ 3 : การเสนอรายงานในรูปกราฟิก

พาคุณสำรวจเทคนิคการวาดแผนภูมิกง แผนภูมิแท่ง และแผนภูมิเคลื่อนไหว

นักปรัชญากล่าวไว้ว่า "ภาพ 1 ภาพ มีค่ากว่าคำพูด 1000 คำ" ถ้าคุณไปเดินชมงานศิลป คุณก็จะเห็นด้วยกับคำกล่าวข้างต้น เมื่อมองในแง่ของการสื่อสาร รูปภาพเป็นการสื่อสารที่ดีและรวดเร็ว จึงทำให้ไมโครคอมพิวเตอร์เข้ามามีบทบาทในการนำเสนอรายงานที่ออกมาในรูปกราฟิก โดยเฉพาะในวงการธุรกิจ การนำเสนอนั้นอาจจะออกมาในรูป กราฟ แผนภูมิแท่ง แผนภูมิกง และรูปภาพ

ในบทความนี้จะแสดงให้เห็นถึงความสามารถของเทอร์โบซีที่จะผลิตกราฟหรือแผนภูมิทางด้านธุรกิจและวิทยาศาสตร์อย่างมีคุณภาพสูง โดยเฉพาะอย่างยิ่งผมจะพาคุณสำรวจเทคนิคการวาดแผนภูมิกง แผนภูมิแท่ง และแผนภูมิเคลื่อนไหว

ชนิดของกราฟิกพื้นฐาน

BGI ได้จัดสรรฟังก์ชันพิเศษสำหรับสร้างกราฟและแผนภูมิ นั่นคือ รูทีนสร้างแผนภูมิกง รูทีนสร้างแผนภูมิแท่ง และการสร้างแผนภูมิแท่งสามมิติ

เริ่มต้น

ถ้าคุณกำลังมองหากฎที่ยากและรวดเร็วในการพัฒนาแผนภูมิและกราฟ คุณก็คงจะหาได้ยากในบทความนี้ รูปภาพแต่ละรูปไม่ได้บอกถึงเนื้อหาอย่างเดียวกันสำหรับบุคคลที่แตกต่างกัน จุดหมายหลักของเราคือการเสนอข่าวสารที่มีอยู่ให้สื่อความหมายไปในทำนองเดียวกัน และทำให้ข่าวสารที่นำเสนอนั้นไม่เป็นเรื่องที่น่าเบื่อหน่าย แต่ให้เร้าใจ จึงต้องเพิ่มสี เพิ่มจุด เข้าไปในกราฟ บทความนี้จึงให้ตัวอย่างง่าย ๆ แต่ให้อิสระแก่คุณในการทดลองดัดแปลงตามแนวความคิดอันหลากหลายที่คุณมีอยู่

รูปที่ 1 โปรแกรม pie.c

```
#include<graphics.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>
#define PI 3.14159
#define torad(d)  (( (double) (d)*PI)/180.0)

double percentvalues[]={9.5,30,40,10.5,10};
char *labels[]={"La-or","Adisak","Sompong","Serm","Kasem"};
char title[]="Monthly Sales";
int numvalues=5;
double aspectratio;
/*The function prototypes for this program*/
void pieceofpie(int x,int y,int radius,double showpercentage,
                                        int color,int fill,char *label);

void showkey(int color,int fill, char *label);
void nextcolorandfill(int *color,int *fill);
main()
{
 int gdriver=CGA,gmode=CGAC3;
 int i,x,y,radius,color,fill,xasp,yasp;

 initgraph(&gdriver,&gmode,"");
 rectangle(0,0,getmaxx(),getmaxy());
 /*write a header to the graph*/
 settextjustify(LEFT_TEXT,TOP_TEXT);
 settextstyle(GOTHIC_FONT,HORIZ_DIR,4);
 outtextxy((getmaxx()-textwidth(title))/2,0,title);
 x=getmaxx()/2; y=getmaxy()/2;
 getaspectratio(&xasp,&yasp);
 aspectratio=(double)xasp/(double)yasp;
 radius=(double)y*5.0/9.0/aspectratio;
 y +=textheight(title)/2;
 x +=6.0/5.0;
 nextcolorandfill(&color,&fill);
 for(i=0;i<numvalues;i++) {
          pieceofpie(x,y,radius,percentvalues[i],color,fill,labels[i]);
          nextcolrandfill(&color,&fill);
 }
 getch();
 closegraph();
 return(0);
 }
 void pieceofpie(int x,int y,int radius,double showpercentage,
                                        int color,int fill,char *label)

  {
          int startangle,endangle,labelangle;
          double labelloc,lradius;
          static double startpercentage=0.0;
          char buffer[40];

  /*offset label radius by a factor of 1.2.This value is dependent on
          *the size of characters you are using
          */
  lradius=(double)radius*1.2 ;
  setfillstyle(fill,color);
  startangle = startpercentage/100.0*360.0;
  endangle  =(startpercentage + showpercentage)/100.0*360.0;
  labelangle=(startangle+endangle)/2;
  labelloc=torad(labelangle);
  pieslice(x,y,startangle,endangle,radius);
   x +=cos(labelloc)*lradius;
   y -=sin(labelloc)*lradius*aspectratio;
```

แผนภูมิกง (pie chart)

ถ้าคุณหยิบวารสารหรือหนังสือพิมพ์ที่วางแผงขายในท้องตลาด คุณต้องพบกับแผนภูมิกงแน่นอน หรือถ้าคุณมองหาซอฟต์แวร์ที่ฮิตติดอันดับเกี่ยวกับการประยุกต์ใช้ด้านกราฟิก ก็คงไม่พ้น Quattro ของบอร์แลนด์ ในตอนนี้เราจะพัฒนายูทิลิตี้ที่สามารถสร้างแผนภูมิกงได้โดยอัตโนมัติ สิ่งที่เราต้องทำคือ สร้างโปรแกรมที่กระทำกับข้อมูลพร้อมกับพิมพ์หัวเรื่องสำหรับแผนภูมิกงได้ โปรแกรมจะจัดขนาดของแผนภูมิกงเองให้เหมาะกับหน้าจอ รหัสต้นฉบับตาม pie.c ดังแสดงในรูปที่ 1

การวาดส่วนย่อยของแผนภูมิกง

ในบทความก่อนเราได้แนะนำฟังก์ชัน pieslice( ) ในตอนนี้เราจะนำเอามาสร้างในแผนภูมิกง ซึ่งในแต่ละส่วนของ slice จะแทนอัตราส่วนของข้อมูลกลุ่มเฉพาะที่เกี่ยวพันกับแผนภูมิทั้งหมด หรือกล่าวอีกแง่หนึ่งคือ slice แต่ละส่วนแทนร้อยละของแผนภูมิกงทั้งหมด การเขียนฟังก์ชันเพื่อสร้างแต่ละส่วนของแผนภูมิกงเราจะใช้ฟังก์ชันชื่อ pieceofpie( ) และจะเรียกใช้ฟังก์ชันนี้เมื่อต้องการจะวาดแต่ละส่วนของแผนภูมิกง ดังถ้อยแถลงในรูปที่ 2

จากรูปที่ 2 อาร์กิวเมนต์ 3 ตัวแรกของ pieceofpie( ) นิยามจุดศูนย์กลางและรัศมีของส่วนของแผนภูมิกงที่จะวาดขึ้น ซึ่งอาร์กิวเมนต์นี้เราคุ้นเคยกันมาแล้วตอนเรียกฟังก์ชัน pieslice( ) แต่เราจำเป็นต้องนำมาใช้ในการวาดขนาดทั้งหมดของ pie แต่ละชิ้น ซึ่งค่าของอาร์กิวเมนต์ขึ้นอยู่กับค่าใน slice-percentage ด้วย ก่อนอื่นลองมาดูวิธีหามุมเริ่มต้นที่จะวาด pie slice ปัจจุบัน

รูปที่ 1 (ต่อ)

```
settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
/*Set text justification depending upon location of pieslice*/
if(labelangle>=300 || labelangle<60)
        settextjustify(LEFT_TEXT,CENTER_TEXT);
else if (labelangle>=60 && labelangle<120)
        settextjustify(CENTER_TEXT,BOTTOM_TEXT);
else if (labelangle>=120 && labelangle<240)
        settextjustify(RIGHT_TEXT,CENTER_TEXT);
else
        settextjustify(CENTER_TEXT,TOP_TEXT);
sprintf(buffer,"%3.11f%%",showpercentage);
outtextxy(x,y,buffer);
startpercentage +=showpercentage;
showkey(color,fill,label);
}
void showkey(int color,int fill,char *label)
{
static int x=3;
static int y=50;

setfillstyle(fill,color);
bar3d(x,y,x+16,y+16*aspectratio,0,0);
settextjustify(LEFT_TEXT,CENTER_TEXT);
settextstyle(SMALL_FONT,HORIZ_DIR,5);
outtextxy(x+20,y+8*aspectratio,label);
y +=18;
}
void nextcolorandfill(int *color,int *fill)
{
static int ffill=1;
static int fcolor =0;
fcolor++;
if (fcolor> getmaxcolor()) {
fcolor =1;
ffill++;
if(ffill>11) {
        closegraph();
        printf("Too many calls to nextcolor...");
        exit(1);
}
}
*fill=ffill;
*color=fcolor;
}
```

```
void pieceofpie (int x, int y, int radius,
        double slicepercentage, int color, int fill)
{
        int startangle, end angle;
        static double startpercentage = 0.0;
        setfillstyle (fill, color);
        startangle = startpercentage/100.0*360.0;
        endangle = (startpercentage + slicepercentage)/100.0*360.0;
        pieslice (x, y, startangle, endangle, radius);
        startpercentage + = slicepercentage;
```

รูปที่ 2 โปรแกรมย่อยของการวาดส่วนของแผนภูมิกง

โดยดูที่จุดเป้าหมายของ static double startpercentage ที่จะเก็บเอาร้อยละทั้งหมดของวงกลมที่แสดงให้บนจอภาพ จึงจำเป็นต้องกำหนดเริ่มต้นของ static double startpercentage ให้เป็นศูนย์ดังนี้

```
static double startpercentage = 0.0;
```

ในแต่ละเวลาที่ pieceofpie( ) ถูกเรียก slicepercentage ก็จะถูกเพิ่มเข้ามา และ pie slice อันต่อไปก็จะเริ่ม ณ ตำแหน่งปัจจุบันนั้น ดังนั้น start-percentage จะบ่งชี้ถึงมุมเริ่มต้นที่จะวาด pie slice ขณะเดียวกัน slicepercentage บอกรูทีนว่าจะสร้างให้แต่ละ slice ใหญ่แค่ไหน แต่ร้อยละนี้ควรจะต้องเปลี่ยนแปลงก่อนที่จะใช้ ซึ่งในเทอร์โบซีฟังก์ชัน pieslice( ) ตัวที่ส่งผ่านไปจะเป็นมุม ไม่ใช่ร้อยละ ดังนั้นจึงต้องแปลงมุมให้เป็นร้อยละด้วยประโยคคำสั่ง

```
startangle = startpercentage/100.0*360.0;
endangle = (startpercentage + slicepercentage)
/100.0*360.0;
```

ประโยคคำสั่งข้างต้น 2 ประโยคคล้ายกัน ซึ่งแต่ละอันขึ้นอยู่กับอัตราส่วนต่อไปนี้

```
มุมที่ต้องการ = ร้อยละ
360 องศา      100 เปอร์เซ็นต์
```

อาร์กิวเมนต์ที่เหลือสองอาร์กิวเมนต์ใน pieceofpie( ) จะเป็นการระบายและการเลือกสี ซึ่งจะเห็นว่า pieceofpie( ) พร้อมที่จะวาดส่วนของแผนภูมิกงได้แล้ว

การเขียนเลเบลของ Pie Slice

การเขียนเลเบลใน Pie Slice มีหลายรูปแบบ แต่ในที่นี้จะเขียนร้อยละของแต่ละ pie slice ไว้ข้าง ๆ เทคนิคนี้จะช่วยให้เห็นขนาดของแต่ละ pie slice เมื่อมองดูฟังก์ชัน pieceofpie( ) สิ่งแรกที่ควรสังเกตคือ ฟังก์ชันนี้จะมีตัวแปรตัวใหม่เรียกว่า lradius ตัวแปร lradius ช่วยกะระยะทางจากจุดศูนย์กลาง

```
Settextstyle (TRIPLEX_FONT, HORIZ_DIR, 1);
if (labelangle > = 300 // labelangle <60)
        settextjustify (LEFT_TEXT, CENTER_TEXT);
else if (labelangle > = 60 && labelangle <120)
        settextjustify (CENTER_TEXT, BOTTOM_TEXT);
else if (labelangle > = 120 && labelangle <240)
        settextjustify (RIGHT_TEXT, CENTER_TEXT);
else
        settextjustify (CENTER_TEXT, TOP_TEXT);
sprintf (buffer, "% 3.11 f%%", showpercentage);
Puttextxy (x, y, buffer);
```

รูปที่ 3
เป็นประโยคคำสั่งในการตั้งค่าพารามิเตอร์ข้อความ

ของแผนภูมิกงไปยังเลเบลที่ได้เขียนขึ้น สำหรับโปรแกรมของเราจะให้ค่า lradius เป็น 1.2 เท่าของรัศมีของแผนภูมิกง ด้วยประโยคคำสั่ง

```
lradius = (double) radius*1.2;
/* เปลี่ยนแปลงไปตามขนาดของตัวอักษร */
```

ซึ่งการคำนวณนี้ขึ้นกับว่ามีที่ว่างเท่าใดบนจอภาพที่จะเขียนเลเบล และความยาวของข้อความที่ใช้

โดยทั่วไปเลเบลที่วางไว้ตรงกลางมุมนี้จะถูกเก็บไว้ใน labelangle และเป็นครึ่งหนึ่งของมุมเริ่มต้นกับมุมสุดท้ายของ pie slice ค่ามุมจะถูกเปลี่ยนให้เป็นเรเดียนโดยอาศัยฟังก์ชันมาโคร torad

```
labelangle = (startangle + endangle) /2;
labelloc = torad (labelangle);
```

ต่อไปโคออร์ดิเนต x และ y ของเลเบลที่ถูกเขียนจะคำนวณ โดยอาศัยประโยคคำสั่ง

```
x+ = cos(labelloc)*lradius;
y− = sin(labelloc)*lradius*aspectratio;
```

ประโยคคำสั่งในรูปที่ 3 เป็นการตั้งค่าพารามิเตอร์ของข้อความ ในรูปแบบที่เป็น triplex stroke font และใช้ if-else เพื่อจัดรูปแบบของเลเบล พร้อมกันนั้นก็ใช้ฟังก์ชัน outtextxy( ) พิมพ์ข้อความออกทางจอภาพ

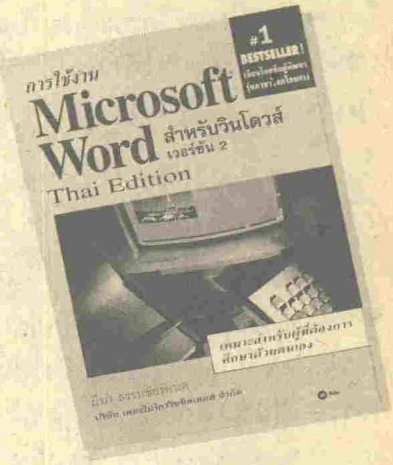
เราได้เห็นวิธีการเขียนชิ้นส่วนแต่ละชิ้นที่ประกอบกันเป็นแผนภูมิกง การเขียนแผนภูมิกงเราก็เพียงแต่เรียก pieceofpie( ) ซ้ำกันหลาย ๆ ครั้งด้วยค่าที่เหมาะสม

การทำให้แต่ละชิ้นเท่ากัน

สิ่งหนึ่งที่น่าจะทำ (ถ้าคุณมีจอสี) ก็คือการให้สีต่าง ๆ กันใน slice แต่ละชิ้น หรืออย่างน้อยที่สุดก็ระบายแต่ละชิ้นด้วยรูปแบบการระบายที่ต่างกัน ในที่นี้จะใช้ฟังก์ชัน nextcolor andfill( ) เพื่อจัดลำดับสีและรูปแบบการระบาย โดยใช้ตัวแปร static เริ่มต้นรูปแบบการระบายด้วย 1 แล้วลงสีด้วย 0

ตรงนี้

สำหรับผู้ที่ต้องการศึกษาด้วยตนเอง



หนังสือเล่มนี้เขียนขึ้นเพื่อตอบสนองความต้องการของผู้ใช้โปรแกรม Word ที่เพิ่มมากขึ้นตั้งแต่ Word ภาษาไทยออกสู่ท้องตลาด เนื้อหาของหนังสือเริ่มตั้งแต่พื้นฐานการใช้งานวินโดว์จนสามารถใช้งานโปรแกรมได้อย่างสมบูรณ์ เนื่องจากมีเนื้อหาครอบคลุมคำสั่งและเทคนิคการใช้ที่สำคัญ เช่น การจัดรูปแบบตัวอักษรและหน้ากระดาษ การสร้างตาราง การแบ่งคอลัมน์และการนำกราฟิกเข้ามาใช้ในการจัดเอกสาร

หาซื้อได้ตามร้านหนังสือชั้นนำทั่วไป
หรือที่ ซีเอ็ดบุ๊คเซ็นเตอร์ทุกสาขา
และสั่งซื้อได้ที่ซีเอ็ด สำนักงานใหญ่
โทร 248-6280

มีจำหน่ายแล้ว

ซีเอ็ด